



MACRO DESIGN AND INTEGRATION

MACRO DESIGN PROCESS

- Top-level macro design
- Subblocks design
- Integrate subblocks
- Macro productization



PROBLEM IN SOC ERA

- Productivity gap
- Time-to-market pressure
- Increasing design complexity
 - HW/SW co-development
 - System-level verification
 - Integration on various levels and areas of expertise
 - Timing closure due to deep submicron

Solution: Platform-based design with reusable IPs




DESIGN FOR REUSE IPS

- Design to maximize the flexibility
 - configurable, parameterizable
- Design for use in multiple technologies
 - synthesis script with a variety of libraries
 - portable for new technologies
- Design with complete verification process
 - robust and verified
- Design verified to a high level of confidence
 - physical prototype, demo system
- Design with complete document set



PARAMETERIZED IP DESIGN

- Why to parameterize IP?
 - Provide flexibility in interface and functionality
 - Facilitate verification
- Parameterizable types
 - Logic/Constant functionality
 - Structural functionality
 - Bit-width, depth of FIFO, regulation and selection of sub- module
 - Design process functionality (mainly in test bench)
 - Test events
 - Events report (what, when and where)
 - Automatic check event
 - Others  (Hardware component Modeling, 1996)



IP GENERATOR/COMPILER

- User specifies
 - Power dissipation, code size, application performance, die size
 - Types, numbers and sizes of functional unit, including processor
 - User-defined instructions.
- Tool generates
 - RTL code, diagnostics and test reference bench
 - Synthesis, P&R scripts
 - Instruction set simulator, C/C++ compiler, assembler, linker, debugger, profiler, initialization and self-test code



LOGIC/CONSTANT FUNCTIONALITY

- Logic Functionality

- Synthesizable code

```
always @(posedge clock) begin if
  (reset==`ResetLevel) begin
  ... end
else begin
  ... end
end
```

- Constant Functionality

- Synthesizable code

```
assign tRC_limit=
  (`RC_CYC > (`RCD_CYC + burst_len)) ?
  `RC_CYC - (`RCD_CYC + burst_len) : 0;
```

- For test bench

```
always #(`T_CLK/2) clock = ~clock;

...
initial begin #(`T_CLK)
  event_1;
#(`T_CLK) event_2;
... end
```



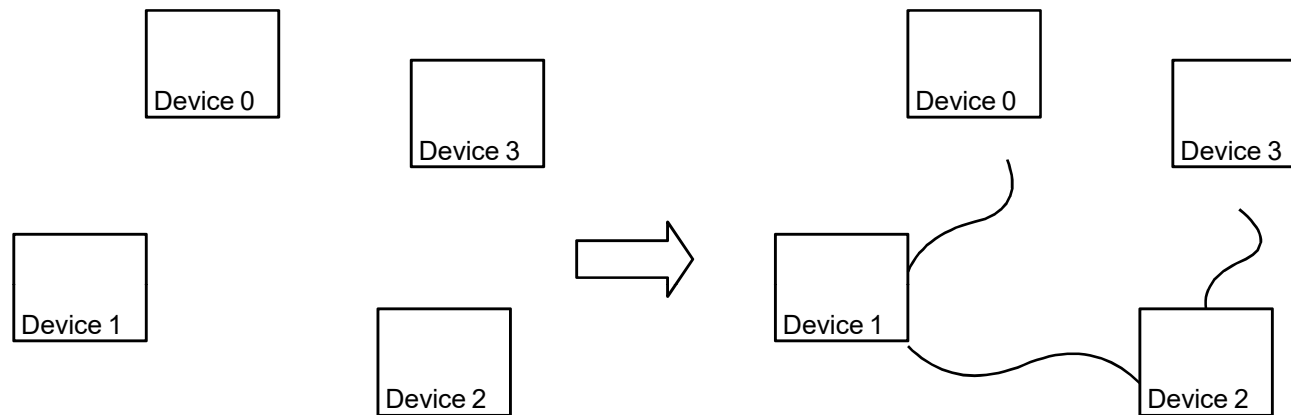
REUSABLE DESIGN - TEST SUITE

- Test events
 - Automatically adjusted when IP design is changed
 - Partition test events to reduce redundant cases when test for all allowable parameter sets at a time
- Debug mode
 - Test for the specific parameter set at a time
 - Test for all allowable parameter sets at a time
 - Test for the specific functionality
 - Step control after the specific time point
- Display mode of automatic checking
 - display[0]: event current under test
 - display[1]: the time error occurs
 - display[2]: expected value and actual value



REUSABLE DESIGN - TEST BENCH

- Use Global Connector to configure desired test bench
 - E.g.: bus topology of IEEE 1394

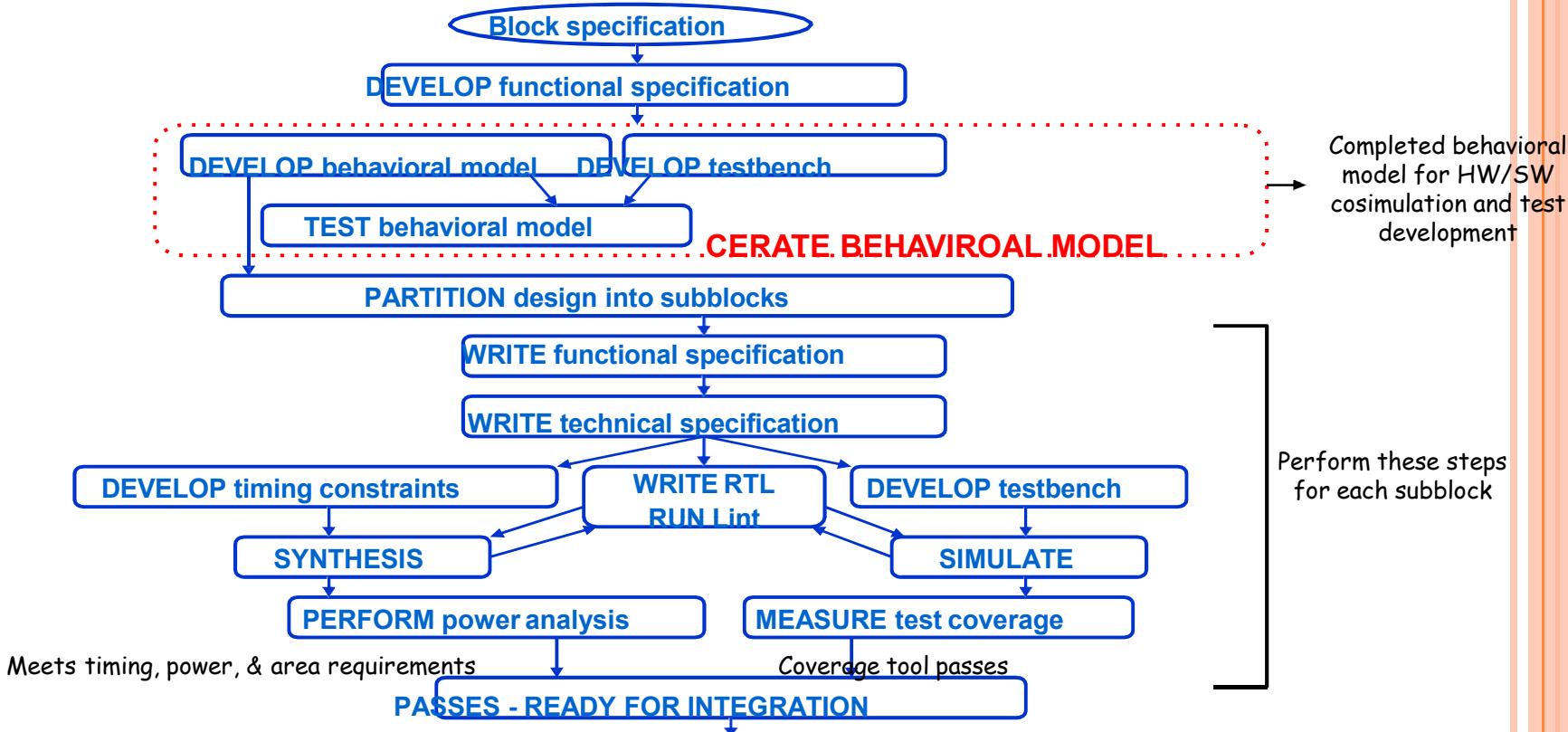


CHARACTERISTICS OF GOOD IP

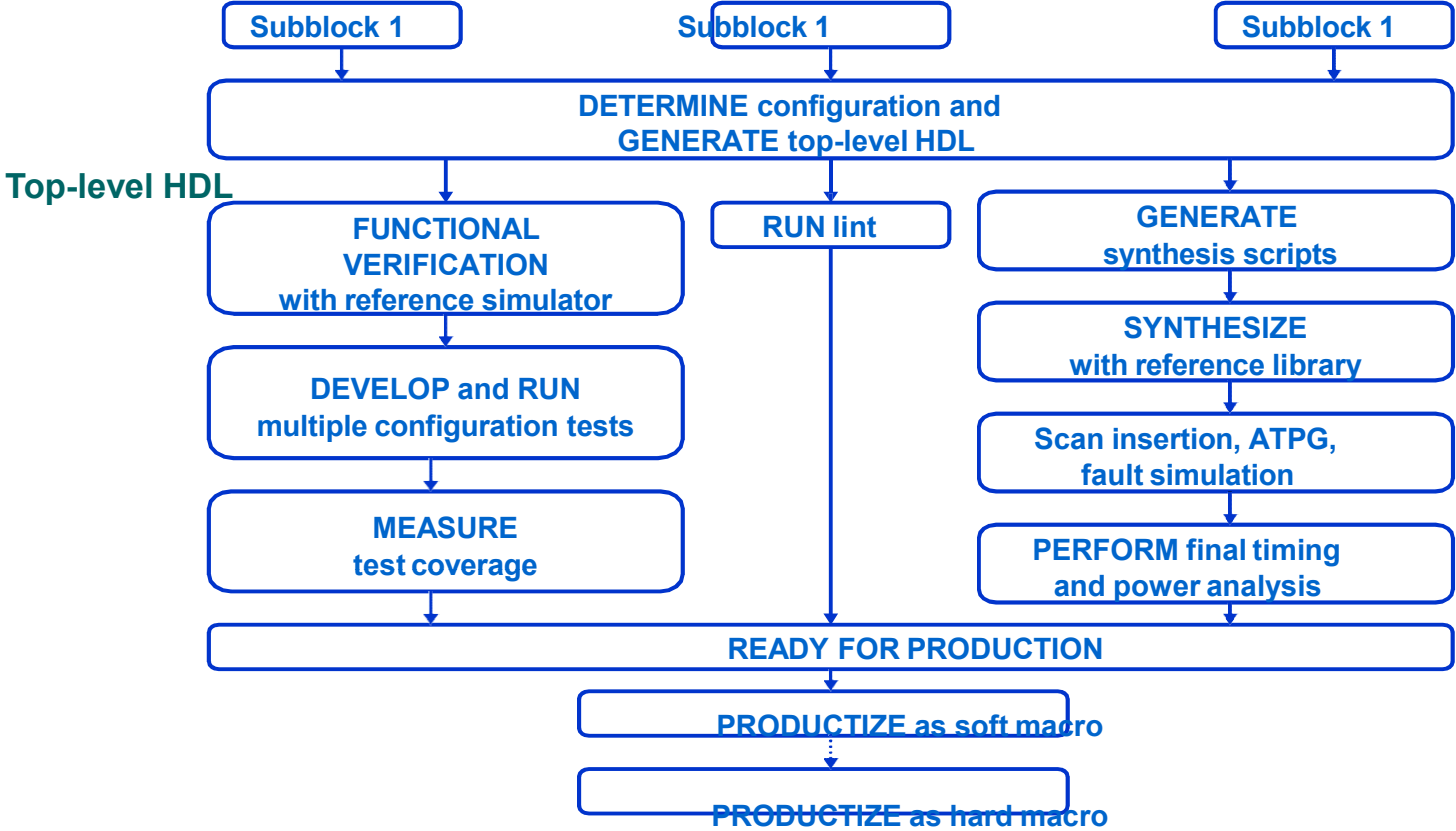
- Configurability
- Standard interface
- Compliance to defensive design practices
- Complete set of deliverables
 - Synthesizable RTL
 - Verification suite
 - Related scripts of EDA tools
 - Documentations



IP CORE MACRO DESIGN PROCESS



MACRO INTEGRATION PROCESS



FOUR MAJOR PHASES

- Design top-level macro
 - macro specification; behavior model
 - macro partition
- Design each subblock
 - specification and design
 - testbench; timing, power check
- Integration subblocks
- Macro productization

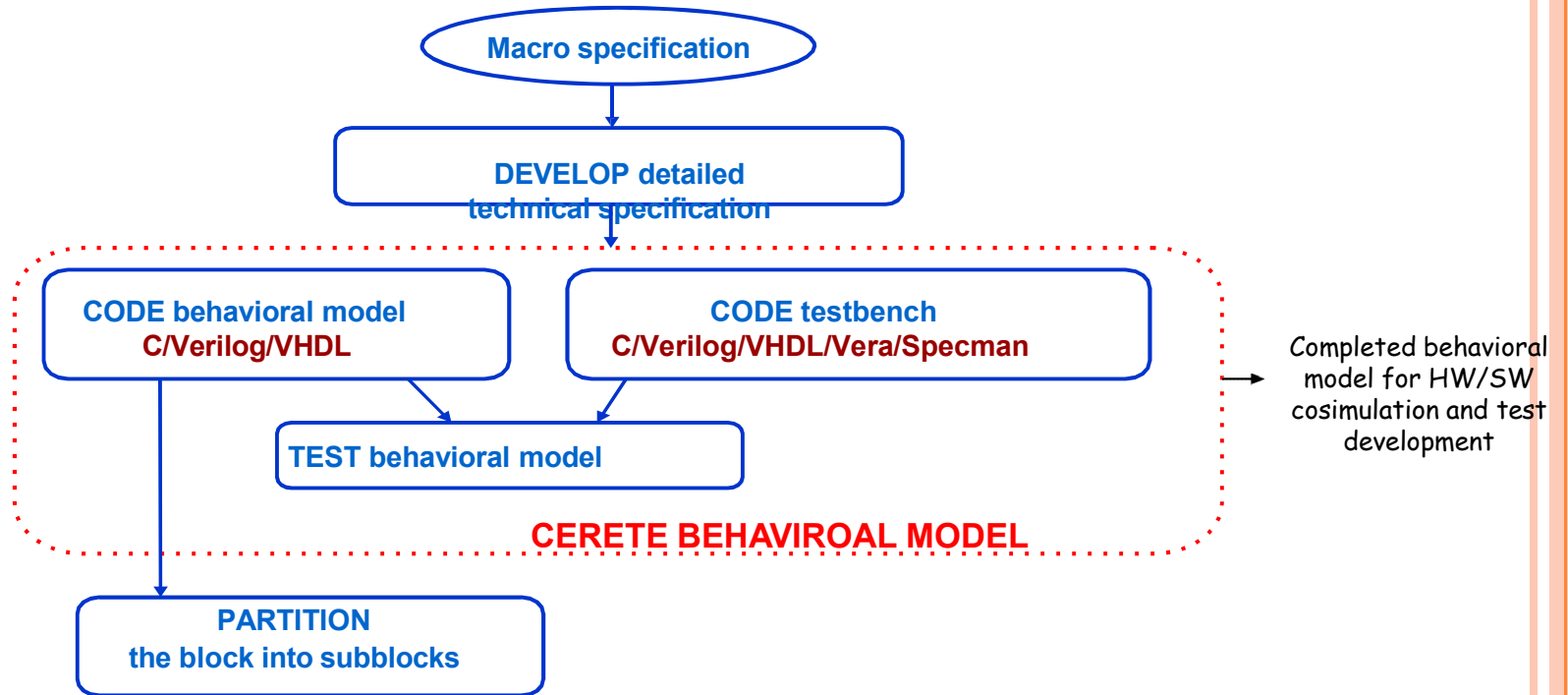


SPECIFICATION AT EVERY LEVEL

- Overview
- Functional requirements
- Physical requirements
- Design requirements
- Block diagram
- Interface to external system
- Manufacturing test methodology
- Software model
- Software requirement
- Deliverables
- Verification



TOP-LEVEL MACRO DESIGN FLOW

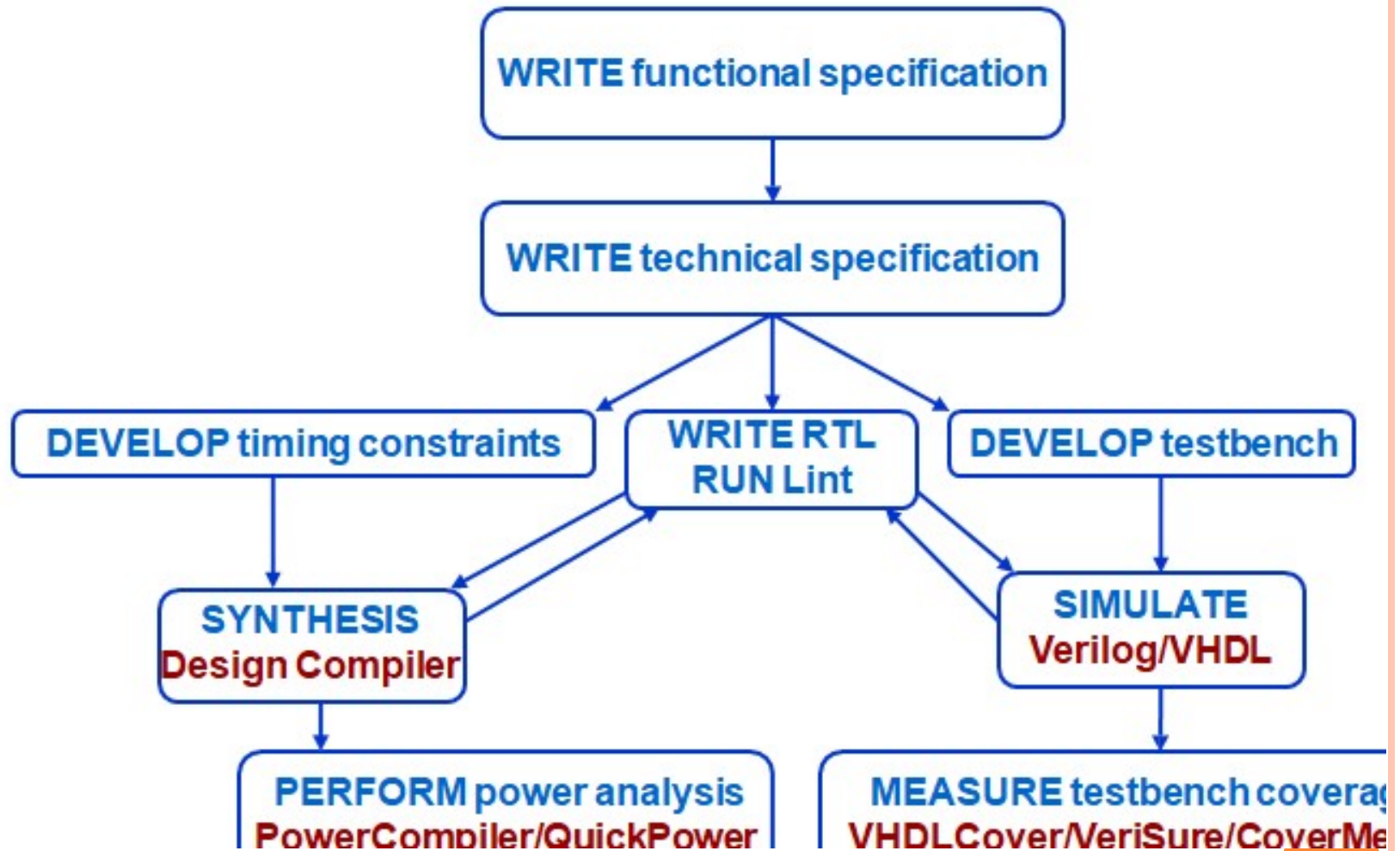


TOP-LEVEL MACRO DESIGN

- Updated macro hardware specification
 - document
- Executable specification
 - language description
 - external signals, timing
 - internal functions, timing
- Behavioral model
 - SystemC, HDL
- Testbench
 - test vector generation, model for under test unit, monitoring and report
- Block partition



SUBBLOCK DESIGN FLOW



SUBBLOCK DESIGN

- Design elements
 - Specification
 - Synthesis script
 - Testbench
 - Verification suite
 - RTL that pass lint and synthesis

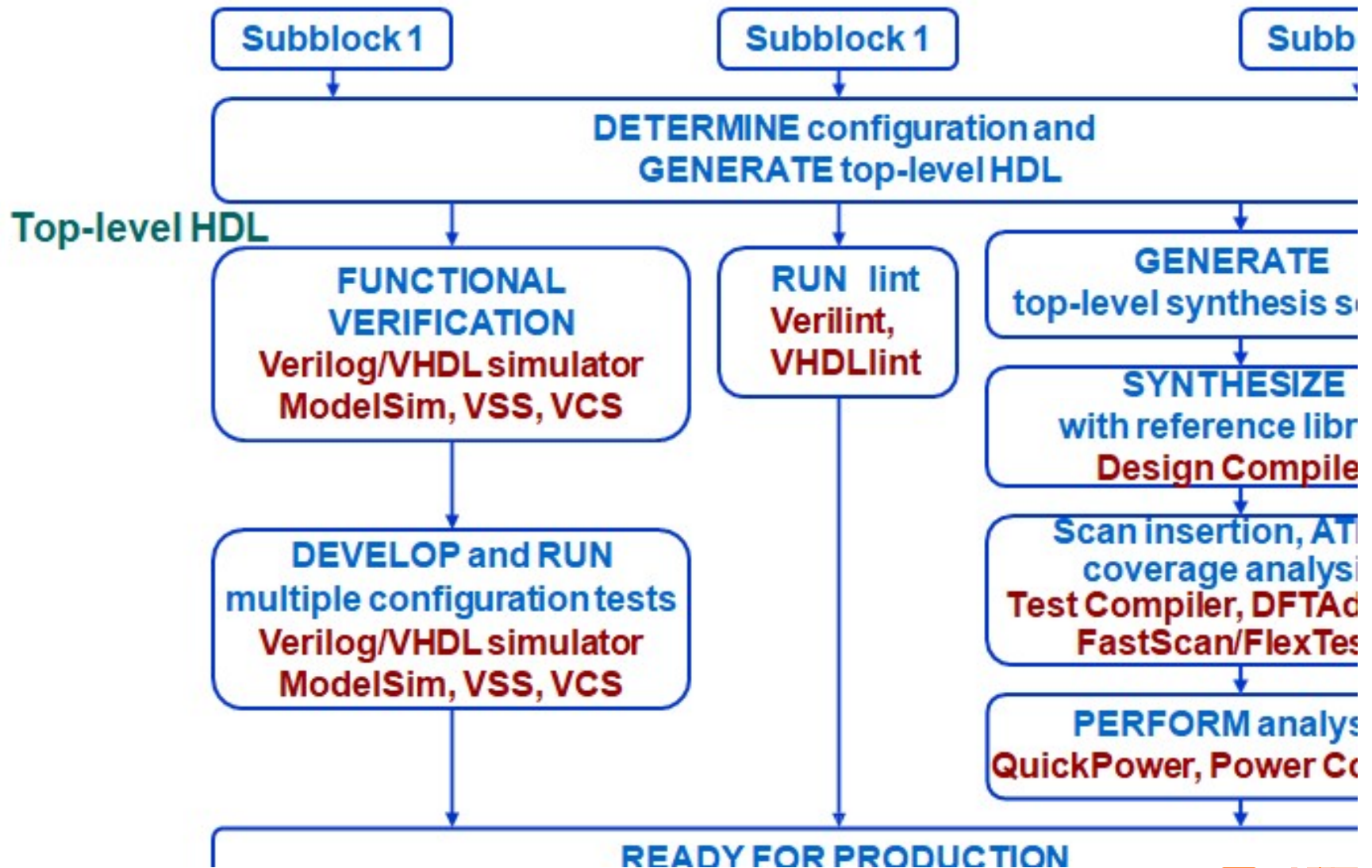


LINTER

- Fast static RTL code checker
 - preprocessor of the synthesizer
 - RTL purification
 - syntax, semantics, simulation
 - timing check
 - testability checks
 - reusability checks
- Shorten design cycle by avoiding lengthy iterations



SUBBLOCK INTEGRATION FLOW

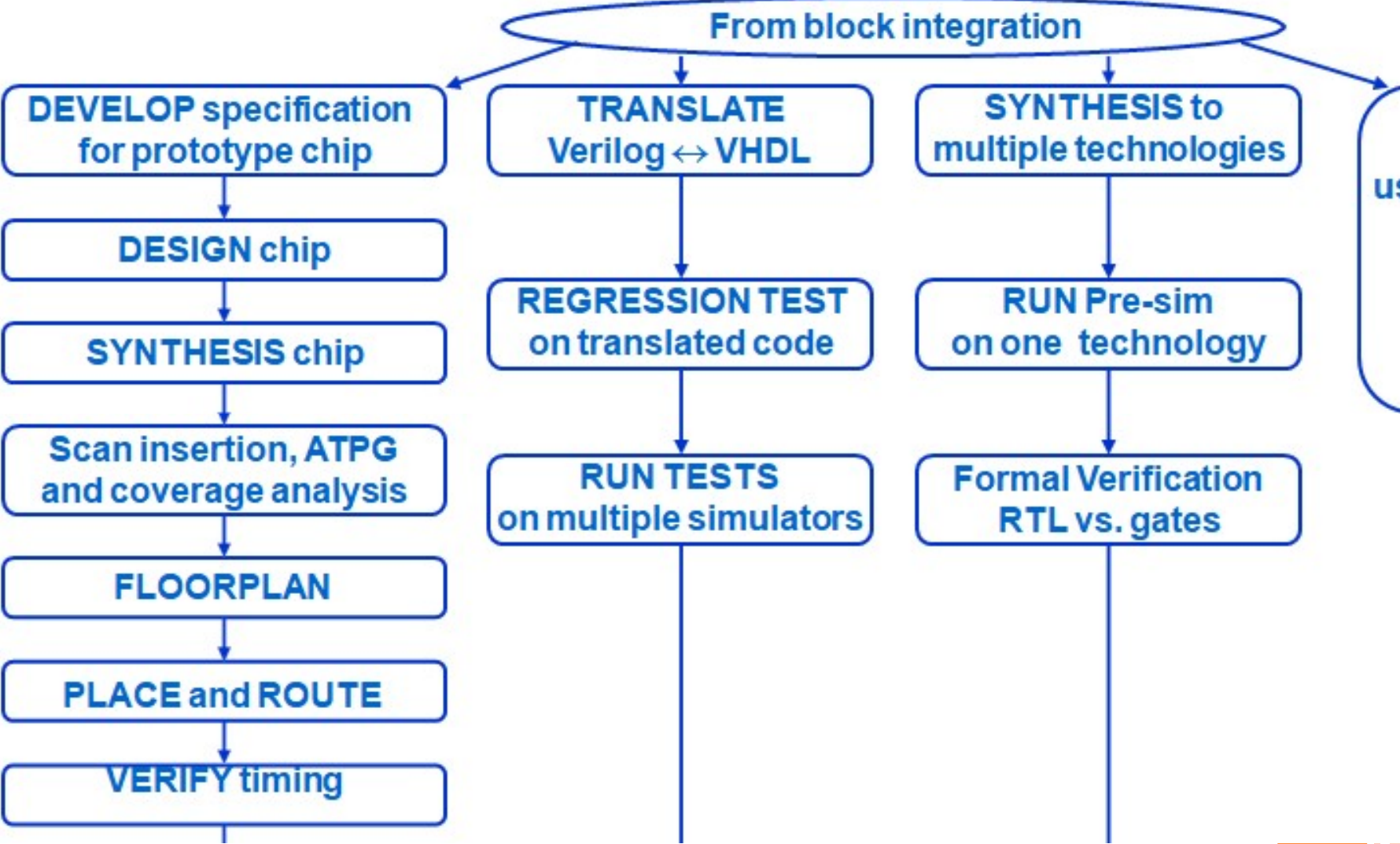


SUBBLOCK INTEGRATION

- Integration process is complete when
 - top-level RTL, synthesis script, testbench complete
 - macro RTL passes all tests
 - macro synthesizes with reference library and meets all timing, power and area criteria
 - macro RTL passes lint and manufacturing test coverage



MACRO PRODUCTIZATION



SOFT MACRO PRODUCTION

- Produce the following components
 - Verilog version of the code, testbenches, and tests
 - Supporting scripts for the design
 - installation script
 - synthesis script
 - Documentation

